REMARKS

In view of the foregoing amendments and following remarks responsive to the Office Action of September 23, 2004, Applicant respectfully requests favorable reconsideration of this application.

In Section 3 of the Office Action, the Office rejected claim 12 under 35 U.S.C. § 112, first paragraph, as failing to comply with the enablement requirement. Specifically, the Office noted that claim 12 recites determining if the first computing entity has previously reflected the requested property. The Office noted that, while a determination is made as to whether the reflection adapter has reflected the property, it is not performed by the first computing entity according to the specification.

The Office's point is well taken. Applicant has amended claim 12 to refer to the reflection adapter rather than the first computing entity in order to correct this clerical error.

The Office further rejected claims 5, 6, and 12 under 35 U.S.C. § 112, fourth paragraph. Particularly, the Office asserted that the recitation in each of these claims of skipping steps recited in claims from which they depend is an improper claiming methodology. Though Applicant respectfully disagrees, Applicant has herein rewritten claims 5, 6, and 12 as independent claims, thus removing this as an issue.

In sections 5-8 of the Office Action, the Office rejected claims 5, 6, and 12 under 35 USC 112, second paragraph, as being indefinite because those claims include non-sequential step numbering.

While Applicant respectfully disagrees with the Office's assertion that this is improper, Applicant has hereinabove amended claims 5, 6, and 12 for unrelated

reasons to make them independent claims, thus also eliminating this issue.

In sections 14-18 of the Office Action, the Office rejected claims 1 and 13-15 under 35 U.S.C. § 102(b) as anticipated by applicant's admitted prior art (AAPA).

Applicant notes a clerical error in the Office Action in that AAPA cannot constitute prior art under 35 U.S.C. § 102(b) and Applicant assumes that the Office intended to refer to 35 USC § 102(a). In any event, Applicant respectfully traverses this rejection. Particularly, the Office asserted that the steps recited in claim 1 are described in Applicant's background section and particularly in steps 104-107 of Figure 1. This is inaccurate.

The present invention is an efficient method for exchanging data between two computer application programs or between a resource library and an application program. In accordance with the invention, XMI documents for transporting the data between the two applications are built on-the-fly rather than being stored in memory. Further, when a resource library or application program receives a request for an object, the resource library creates the resource to which that object corresponds, but does not populate that resource. Next, the resource library populates the resource with only the object(s) requested. Then the resource is returned to the requesting application program.

This is very different from the prior art described on page 7 of the present application and shown in the flow chart of Figure 1. In the flow chart shown in Figure 1, the resource factory does not create the resources on-the-fly, nor does it populate the resource with only the object requested. Rather, it stores fully populated resources in memory. It retrieves them when an object in that resource is requested and sends the entire resource to the requesting application program, even if only a single object in that resource was requested. This is wasteful and inefficient.

The present invention solves that and other problems.

In any event, it should be apparent that the prior art described hereinabove and in the background section of the present application does not, in fact, include a resource factory "including a plurality of software modules for building resources from a data source". Furthermore, it does not incorporate the steps of "responsive to a request for an object from a first computing entity, selecting a software module for building a resource of the type to which said requested object corresponds", "building a resource for containing said requested object..., said resource populated with information defining said resource, but not containing said requested object", or "inserting said requested object into said resource".

The resource factory described in the background section of the present application does not build the resources. It retrieves the already built resources from a data store. Accordingly, independent claim 1 clearly distinguishes over AAPA. All other of the claims rejected over AAPA depend from claim 1, and therefore, distinguish over AAPA for at least the same reasons.

In sections 18-34 of the Office Action, the Office rejected claims 1-4, 7-9, 11, and 13-15 as anticipated by Dorsett under of U.S.C. 102(e), claim 19 [sic, 10?] under 35 U.S.C. 103(a) as obvious over Dorsett in view of Francis, and claims 5, 6, and 12 under 35 U.S.C. 103(a) as obvious over Dorsett in view of Kumar. Applicant notes that there is no claim 19 in this application and assumes that the Office intended to refer to claim 10.

In any event, Dorsett has an effective prior art date of January 5, 2001, only approximately three months prior to the filing date of the present application. In fact, Applicant had reduced the present invention to practice prior to January 5, 2001. Accordingly, Dorsett, the Office's primary reference in all of the prior art

rejections (other than the ones that rely on AAPA), is not, in fact, prior art.

Applicant can provide a suitable declaration and evidence of such facts should it

become necessary to remove Dorsett from consideration.

However, that should not be necessary because Dorsett does not teach that

for which it has been cited, in any event. Particularly, Dorsett pertains to a

computer program for processing data from a chemical experiment. The process

includes receiving data from a chemical experiment, in which experiment a plurality

of chemical compounds (i.e., materials or members) are mixed in all possible

permutations, and then generating a representation of the results of all of the

permutations. The representation includes data defining an experiment object

having a plurality of properties derived from the chemical experiment. The

experiment object is associated with the library of materials. The representation

also includes data defining one more element objects. Each element object is

associated with one or more members of the library of materials. A data model and

corresponding data structures for describing such experiments are disclosed.

The Office is relying on paragraphs 83-88 of Dorsett as teaching the

elements of claim 1. This portion of Dorsett describes how data about the

chemistry experiment can be retrieved by database server process 130 from

database 180 and presented to the user for viewing. While Dorsett does talk about

parsing Java objects into and out of XML, it does this using a single, direct, and

well-known mechanism that is substantially different from the present invention.

In the present invention, a plurality of software modules for building

resources is maintained, each software module designed to build a resource of a

particular type. Then, responsive to a request for an object of a particular resource

type, the appropriate software module for building resources of that type is selected

to build that resource. Then, the requested object is inserted into the resource and

returned to the requesting program.

Dorsett discloses nothing like this. In Dorsett, there is no issue as to different types of resources, i.e., different data formats preferred by a given requesting program. In Dorsett, database server process 130 retrieves data from database 180 using a method public String GetObject2. (Dorsett, paragraph 83.) This method accepts a string containing the name of the object to be retrieved and 2 Boolean parameters that control retrieval of data. (Dorsett, paragraph 84.)

As described in paragraph 85 of Dorsett, the overall process comprises retrieving the object by an ID from the database and creating a Java object by name and populating its fields. This Java object is then mapped into an XML document and the document is returned to the requester as a string using the WriteXMLNode method.

This is an old and well known technique and has virtually nothing to do with what Dorsett considers to be his innovation.

The process described in Dorsett is quite similar to that which is described in the Background of the Invention section of the present application. In short, Dorsett discloses nothing more than (1) retrieving the requested object from the database, (2) populating the fields of a Java object with that data, (3) mapping it into an XML document, and (4) transmitting the XML document.

Thus, referring to claim 1, Dorsett does not have "a resource factory for building resources, <u>said factory including a plurality of software modules for building resources from a data source, each said software module designed to build a resource of a particular type</u>".

Claim 1 also recites, (1) building a resource, (2) inserting an object into the resource, and (3) transmitting the resource using a transport mechanism (which, in a preferred embodiment as described in later dependent claims, is XML document).

Regardless of how broadly the Office tries to interpret the terms "resource", "object" and "transport mechanism" in claim 1, there is at least one step of claim 1 missing from Dorsett. Dorsett has only two elements (namely, the Java object and the XML document) that need to be read on three claim elements (namely, the resource, the object, and the transport mechanism).

Interpreting the language of claim 1 as broadly as possible, the object recited in claim 1 can be the Java object in Dorsett and the transport mechanism recited in claim 1 can be the XML document in Dorsett. However, that leaves nothing in Dorsett that can correspond to the resource.

The Office appears to asserting that the Java object corresponds to the recited resource as well as the recited object. However, a Java object is not a resource; it is an object. Furthermore, it is improper to read the Java object on both the claimed "resource" and the claimed "object". Such a reading makes no sense in the context of the claim.

Even further, it appears that the Office is reading Dorsett's XML document on the claimed resource as well as the claimed transport mechanism. Again, this is clearly improper for at least two reasons. First, an XML document is not a resource; it is a transport mechanism. Furthermore, even if it could somehow be deemed a resource, it is improper to consider it both the claimed resource and the claimed transport mechanism. Such a reading makes no sense in the context of claim 1.

Dorsett simply has no concern for the problems that the present invention solves. Dorsett is merely retrieving an experiment to display to a user where there is no issue about different users requiring the information in different formats. There is no issue about whether or not to populate the resource with all of the corresponding objects or just the requested object.

Dependent claim 3 even further distinguishes over Dorsett. Dependent claim 3 adds the additional steps of: providing a reflection adapter factory for populating objects within resources, each module designed for an environment corresponding to a requested object; responsive to a request for a property of that object, selecting the appropriate reflection adapter for the environment of the particular requested property; populating the object with the requested property; and providing the requested property to the first computing unit.

The Office asserted that this is disclosed on page 9 of Dorsett, but there is nothing on page 9 even remotely resembling these features.

With respect to dependent claim 9, the Office asserted that Dorsett discloses using XML documents, which is a superset of XMI documents and, therefore, reads on claim 9. This is clearly an improper rejection. However, first please note that applicant has herein amended the form of claims 7 and 9 in that there was no antecedent basis for the term "said files". Applicant has rewritten those claims to recite that the transport mechanism itself is the XML document or XMI document. This eliminates the antecedent basis issue and is also more consistent with the specification as originally filed.

In any event, the Office's position that the disclosure of a superset of a claim limitation constitutes a disclosure of the claim limitation is erroneous. In essence, the Office is asserting that disclosure of a genus constitutes disclosure of a species. This obviously is incorrect. Accordingly, dependent claim 9 further distinguishes over the prior art.

With respect to the rejections of claims 5, 6, 10, and 12 as obvious over Dorsett in view of the particular secondary reference, those rejections are overcome, at least by virtue of the fact that Dorsett does not disclose of the

limitations of the independent claims discussed above, all of which are incorporated either literally or by dependence in claims 5, 6, 10, and 12. The secondary references do not provide the above-discussed teachings lacking from the primary reference, Dorsett.

In reviewing the present application for purposes of preparing this response, Applicant discovered several typographical and clerical errors in the specification, which it has corrected herein.

Furthermore, in reviewing the claims for purposes of preparing this response, Applicant noted that the term "requested" as a modifier to some of the nouns, such as "object" and "property" was unnecessary because the claims only refer to one "object" or "property". Accordingly, Applicant has cleaned up the claim language to eliminate unnecessary verbiage.

In view of the foregoing amendments and remarks, this application is now in condition for allowance. Applicant respectfully requests the Examiner to issue a Notice or Allowance at the earliest possible date. The Examiner is invited to contact the Applicant's undersigned counsel by telephone call in order to further the prosecution of this case in any way.

Respectfully submitted,

Theodore Naccarella
Registration No. 33,023
Synnestvedt & Lechner LLP
2600 Aramark Tower
1101 Market Street
Philadelphia, PA 19107
Telephone: (215) 923-4466
Facsimile: (215) 923-2189
Attorney for Applicant

M:\TNaccarella\CLIENTS\IBM\24929\Patent Office\app-trans